

# HOW IT WORKS

## The BIOS

Your PC has a program built into its hardware long before you insert the Windows installation disc, but what exactly does it do? Mike James explains the role of the BIOS – and why its days are numbered

**T**he basic input/output system (BIOS) has been with us since the PC was invented in 1981, but it is still misunderstood. If you've ever tinkered with, cursed at or flashed a BIOS, now is the time to find out what you're dealing with.

The BIOS was built into the first PC as a way of getting the machine started, configuring it and smoothing out the differences between variations of PC hardware. It was common for computers to have the OS in read-only memory (ROM). This meant the machines were ready to use as soon as they were switched on because the OS was already loaded. However, loading software was difficult and time consuming because disk drives were expensive and so not common. Most machines had in their ROM a BASIC interpreter and a mini-operating system offering routines for simple tasks. When the IBM PC was designed, it followed the same plan. Its ROM contained Microsoft BASIC so it could be used without a disk drive. It also had a BIOS ROM, developed from a prevalent operating system of the period: CP/M. Over time the ROM BASIC became irrelevant and was dropped, but the BIOS ROM is still with us.

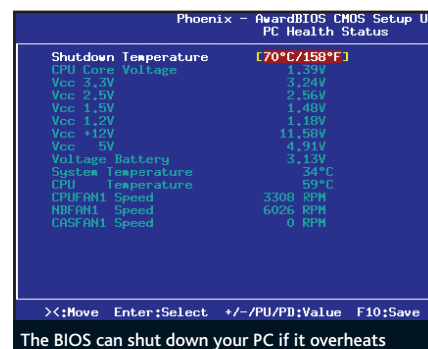
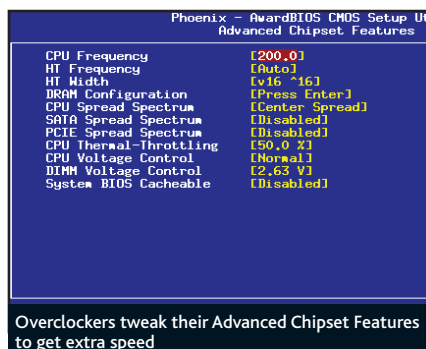
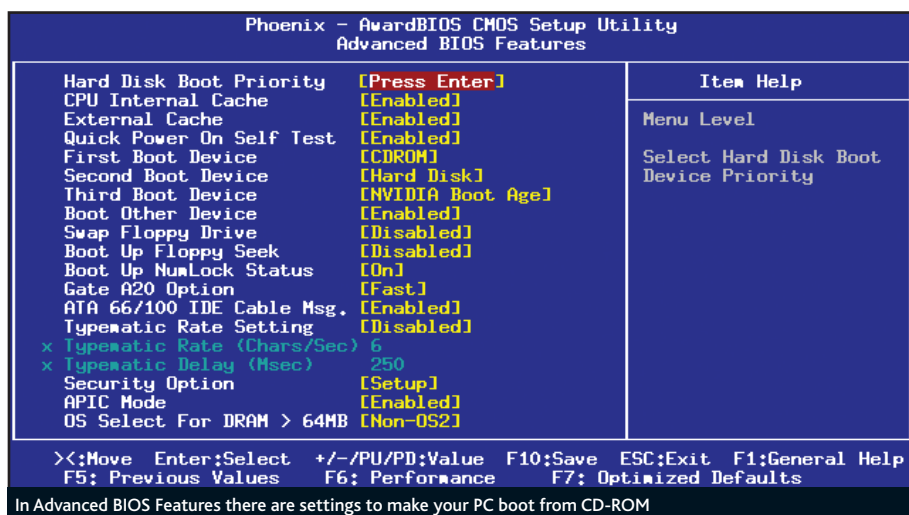
So what does the BIOS do that is so important? Well, it performs four separate tasks:

- 1 **Testing:** When you switch on your PC the BIOS performs the power-on self-test (POST) to make sure the core components of your PC are working.
- 2 **Configuration:** The BIOS specifies appropriate settings for all components on the motherboard.
- 3 **Booting up:** The BIOS loads your operating system from the hard disk.
- 4 **Hardware access:** The BIOS also provides access to some of your PC's components and features. The mechanism by which the BIOS is invoked to access something is an interrupt call.

The BIOS performs the first three tasks when you switch a machine on. The hardware access function is used once the machine starts and the operating system is in control.

### SELF-STARTER

Starting the machine, or booting it, has always been the BIOS's most important job. When you switch on the PC, its RAM is full of random bits. Before it can do anything useful, it must load an operating system; the BIOS tells it how to do this. The BIOS boot code is stored in ROM so it's available as soon as the machine is turned on. The PC's hardware is designed to run the boot code stored in a specific point in the ROM automatically when switched on. The boot code then performs all the jobs necessary to get the machine going, including loading the operating system from the hard disk.



Additional hardware that you plug into expansion sockets on your motherboard can add to the BIOS, giving you the flexibility to expand your PC. Graphics cards carry a Video BIOS, which the main BIOS ROM detects at startup and adds to the startup routine. You'll often see your graphics card reported when you boot your PC. BIOS extensions have also been used to enable a PC to boot from a disc located on another PC on a network.

One of the first stages in the boot process is the POST, which tests such things as the memory. However, there is usually an option to skip this or make it more intensive, depending on how long you are prepared to wait for your PC to start up. POST tests can take a long time if they are rigorous enough to be useful. The original PC BIOS had a system of beep codes that told the user if anything was wrong; a Morse Code-like sequence of beeps from the PC speaker enabled the BIOS to report an error even if the display wasn't working. Many BIOS ROMs still do this, but you may get a cryptic error message on the screen and error LEDs may light up on the motherboard.

The POST routine detects severe faults such as a lack of memory, keyboard or video hardware. This is useful when building or upgrading a PC, but when your PC is running properly it's wise to switch off POST to enjoy faster booting. POST rarely ever spots subtle or developing faults.

### GO CONFIGURE

Once the POST has completed successfully the BIOS configures the motherboard hardware. It makes all kinds of settings, from adjusting the clock to setting the speed at which your memory is accessed. To do this, it uses the defaults that the manufacturer programmed in or the settings you entered using the BIOS Setup utility.

This utility lists all the settings you can tweak. You make it appear by holding down a combination of keys when you first switch on the machine. Use the Delete key to enter most AMI and Award BIOS utilities. If this fails try F1, F2 or F10. The BIOS Setup isn't difficult to use, but some of the settings are very technical. For instance, changing your memory timings is highly complicated. While it can speed up



a machine, it can also make it unreliable if the memory fitted isn't fast enough. So change BIOS settings with care and one at a time, noting the original settings so you can retrace your steps. The best-known BIOS settings guides on the internet are out of date; it's better to type the name of the specific setting you want (along with the keywords 'BIOS' and 'setting') into Google.

The role of the BIOS in configuring the system changed dramatically with the introduction of Plug and Play (PnP) in 1995. Using PnP, the BIOS was meant to take care of configuring all the hardware attached to a PC. The BIOS kept a table of the resources needed and assigned them as necessary. In theory, a PnP BIOS should configure a machine automatically when you switch it on; manual configuration should be unnecessary. But the BIOS configuration is not perfect, and recent versions of Windows take over the management of PnP devices when the operating system loads, anyway.

Once the system is configured, the boot process starts. The BIOS has identified which potential boot devices are connected during configuration. The BIOS may ask if you want to boot from hard disk or CD-ROM – or it may just try the First Boot Device as specified in the Advanced BIOS Features page of the BIOS Setup utility.

Booting reads a program, the loader, from a standard area of the boot device, the master boot record (MBR). Once the loader is stored in memory it loads and initialises the operating system. The role of the BIOS in starting the machine is over.

## SORRY TO INTERRUPT

The second role of the BIOS is to provide access to various hardware items and features. This is why it was initially called the basic input/output system. For example, the BIOS includes a small routine that handles the keyboard. The operating system calls this routine to find out which keys you are pressing. You may wonder why the operating system doesn't just go straight to the keyboard hardware. It could, but then any changes to the hardware would necessitate a change to the operating system. It's easier for the motherboard manufacturer to change its BIOS to deal with the new keyboard controller it has fitted. In effect, the BIOS isolates the operating system from hardware changes, so no matter what keyboard hardware you use, for example, the operating system can call the same BIOS routine to read it.

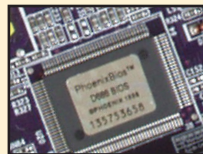
The BIOS routines take the form of software interrupts, which are accessed from fixed locations in the PC's memory so the operating system always knows where to look. To deal with the keyboard, the operating system uses Int 16 to run the appropriate BIOS code.

## The clean room clones

**W**hen IBM designed the PC it made its design 'open'. It used off-the-shelf parts. You could even buy a circuit diagram of the entire machine. What stopped other companies creating IBM PC clones was the BIOS code: IBM owned the copyright. Without your own BIOS, you couldn't create a clone. Copying the BIOS, even in part, would result in a lawsuit from IBM. To prove that their BIOSes were legal, early pioneers such as

## Flashing the BIOS

**B**IOS code is often buggy, and even the best code can struggle to cope with new hardware when it arrives. To alleviate this problem, modern machines store their BIOS code in re-writable Flash erasable programmable read-only memory or EPROM. If you can find an updated or debugged version of the BIOS code for your machine, you can download it and use a special utility to transfer the new code to the BIOS chip.



You may be rewarded with improved stability or support for faster processors.

This process, called **Flashing the BIOS**, isn't without risk. Any interruption in the transfer, such as a power cut, can leave your machine without a BIOS and unable to start. Some motherboards have special features to help your PC recover from this dire situation, but in their absence you have no choice but to find and fit a new BIOS chip that already has the code in it.

The main purpose of the BIOS code is to provide a uniform interface for varying hardware so the operating system always works. In many cases, however, manufacturers have used the BIOS code to add features. For example, many portable computers have special function keys that are not found on standard keyboards. These control such things as screen brightness, contrast, sound volume and so on. The operating system doesn't know anything about these non-standard keys; they are dealt with by suitable BIOS modules from the manufacturer.

## THE MODERN BIOS

The BIOS is old. Although PC hardware has moved through several generations, the BIOS has remained unchanged. Most of it is written using out-dated 16-bit processor instructions. It's also a mess. Each time a new piece of hardware is introduced, the BIOS is modified to take account of it, and the same is true for new technologies such as PnP and power management. As a result, the BIOS often causes problems for users. In modern motherboards the BIOS can usually be upgraded – see our 'Flashing the BIOS' box above. What's more, although the BIOS still has a role to play in getting the machine started with the POST, configuration and boot up, its input/output routine – the software interrupts – is diminishing. For example, the Linux operating system makes no BIOS calls while it is running because it knows about all the standard hardware it is likely to encounter and deals with it directly. Linux uses device drivers to deal with any special non-standard hardware.

Windows has also reduced its use of the BIOS while it runs, although information on what it does

is hard to come by. Some research by programmers designing a version of Linux that boots without a BIOS has discovered that only a small number of

BIOS calls are made while

Windows 2000 and XP are running. The BIOS32 extension, which was introduced to support 32-bit processors, is completely ignored. Even the PnP and power management parts of the BIOS are used only while the system boots and are then similarly ignored.

In short, the BIOS has been relegated to

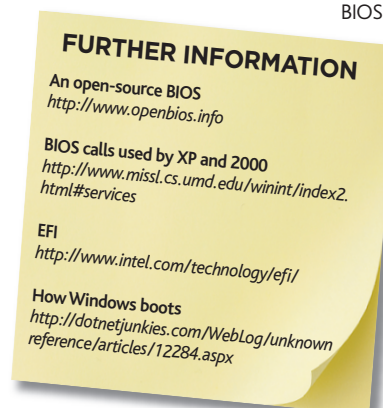
the role of getting the machine started.

Only a small number of its facilities are used while Windows is running. All the hardware-specific code has been moved into Windows drivers. Windows needs the BIOS to get started, but then it tries to do things on its own.

## MOVING ON

As the BIOS is old and less important than it used to be, there are moves to replace it with something better. There is an open-source attempt to build a better BIOS that would be free to use, and another to turn a portion of the Linux kernel into a BIOS that will boot the system. In both cases, the role of the BIOS is to configure the machine, start it up and get out of the way.

As long as the existing BIOS is 16-bit, even the most modern processors have to pretend to be ancient 16-bit 8086 chips when using the BIOS to start up. This is quite annoying for processor manufacturers so Intel, with the support of companies such as Microsoft, has proposed a BIOS replacement called Extensible Firmware Interface (EFI). This is supported by 64-bit Windows and Linux. The EFI is a modular BIOS based on drivers written in C and it supports booting in 64-bit mode. It is simply the new and better BIOS we need.



### FURTHER INFORMATION

An open-source BIOS  
<http://www.openbios.info>

BIOS calls used by XP and 2000  
<http://www.missl.cs.umd.edu/winint/index2.html#services>

EFI  
<http://www.intel.com/technology/efi/>

How Windows boots  
<http://dotnetjunkies.com/WebLog/unknown/reference/articles/12284.aspx>

## NEXT MONTH

### VPNs

We explain how you can remote access your network without compromising security